PRACTICA M 42

1. Mediante el uso de Python y SQLite genera un archivo de base de datos denominado "RH.db" que contenga una tabla bajo el nombre de "Detalle" que importe la información proveniente del archivo CSV mostrada en la imagen previa. Mediante diversas consultas en SQL:

In [2]:
```
import sqlite3

conn = sqlite3.connect('RH.db')
```

In [4]:
```
import os
import csv

os.chdir('C:/Users/Isaac/Desktop/IHD/EBAC DT/CIENCIA DE DATOS/M42 DS')
with open('recursos_humanos.csv') as f:
    reader = csv.reader(f)
    data = list(reader)
```

In [6]:
```
cur = conn.cursor()
```

2. Despliega los campos "sales", "salary" y "satisfaction_level" para todos los registros de la base de datos de forma que los registros se encuentren ordenados de manera descendente de acuerdo al campo "satisfaction_level".

In [7]:
```
cur.execute("""
    SELECT Sales, Salary, Satisfaction_level
    FROM Detalle
    ORDER BY Satisfaction_level DESC;
""")
res = cur.fetchall()
res
```

Out[7]:
```
[('technical', 'low', 1.0),
 ('technical', 'low', 1.0),
 ('support', 'low', 1.0),
 ('sales', 'low', 1.0),
 ('technical', 'low', 1.0),
 ('marketing', 'high', 1.0),
 ('hr', 'low', 1.0),
 ('support', 'low', 1.0),
 ('technical', 'low', 1.0),
 ('sales', 'medium', 1.0),
```

3. Visualiza los registros para los campos "salary", "number_project" y "satisfaction_level" de manera que sean ordenados en primera instancia de forma ascendente por "number_project" y en segundo lugar de forma descendente por el campo "satisfaction_level"

In [14]:
```
cur.execute("""
    SELECT Salary, Number_Project, Satisfaction_Level
    FROM Detalle
    ORDER BY Number_Project ASC, Satisfaction_Level DESC;
""")
res = cur.fetchall()
res
```

Out[14]:
```
[('low', 2.0, 1.0),
 ('medium', 2.0, 1.0),
 ('medium', 2.0, 1.0),
 ('medium', 2.0, 1.0),
 ('medium', 2.0, 1.0),
 ('medium', 2.0, 1.0),
 ('medium', 2.0, 0.99),
 ('low', 2.0, 0.99),
 ('medium', 2.0, 0.99),
 ('low', 2.0, 0.98),
 ('high', 2.0, 0.98),
 ('low', 2.0, 0.98),
 ('low', 2.0, 0.98),
```

4 Muestra el promedio obtenido de la última evaluación (campo "last_evaluation") para cada departamento (campo "sales"). *Sugerencia: Use la función "AVG"

In [9]:
```
cur.execute("""
    SELECT Sales, AVG(Last_evaluation)
    FROM Detalle
    GROUP BY Sales;
""")
res = cur.fetchall()
res
```

Out[9]:
```
[('IT', 0.716829665851672),
 ('RandD', 0.7121219822109279),
 ('accounting', 0.7177183833116036),
 ('hr', 0.7088497970230044),
 ('management', 0.7240000000000004),
 ('marketing', 0.7158857808857806),
 ('product_mng', 0.7147560975609766),
 ('sales', 0.7097173913043466),
 ('support', 0.7231090174966335),
 ('technical', 0.7210992647058838)]
```

5 Resuelve nuevamente el punto anterior pero agrupando sus resultados por el departamento (campo "sales") y salario (campo "salary"), mostrando también el campo "Salary".

In [11]:
```
cur.execute("""
    SELECT Sales, Salary, AVG(Last_evaluation)
    FROM Detalle
    GROUP BY Sales, Salary
""")
res = cur.fetchall()
res
```

Out[11]:
```
[('IT', 'high', 0.7166265060240965),
 ('IT', 'low', 0.7156650246305413),
 ('IT', 'medium', 0.7181869158878511),
 ('RandD', 'high', 0.7005882352941177),
 ('RandD', 'low', 0.7141758241758245),
 ('RandD', 'medium', 0.7116935483870968),
 ('accounting', 'high', 0.7245945945945944),
 ('accounting', 'low', 0.7138826815642458),
 ('accounting', 'medium', 0.720298507462687),
 ('hr', 'high', 0.7437777777777778),
 ('hr', 'low', 0.717820895522388),
 ('hr', 'medium', 0.6961002785515316),
 ('management', 'high', 0.7158222222222224),
 ('management', 'low', 0.7128333333333329),
 ('management', 'medium', 0.7411111111111105),
```

6 Extrae aquellos departamentos para los cuales el promedio de accidentes de trabajo sea mayor a 0.15. Para dicho efecto muestre el campo "sales" y el promedio de "Work_accident". ¿Cómo cambiaría su consulta si lo que quisiera mostrar es aquellos departamentos donde el número de accidentes sea mayor que 200?

In [13]:
```
cur.execute("""
    SELECT Sales, AVG(Work_accident)
    FROM Detalle
    GROUP BY Sales
    HAVING COUNT(Work_accident) > 0.15;
""")
res= cur.fetchall()
res
```

Out[13]:
```
[('IT', 0.1336593317033415),
 ('RandD', 0.17026683608640406),
 ('accounting', 0.12516297262059975),
 ('hr', 0.12043301759133965),
 ('management', 0.1634920634920635),
 ('marketing', 0.16083916083916083),
 ('product_mng', 0.14634146341463414),
 ('sales', 0.14178743961352658),
 ('support', 0.15477792732166892),
 ('technical', 0.1400735294117647)]
```

In [15]:
```
#  ¿Cómo cambiaría su consulta si lo que quisiera mostrar es aquellos departamentos
#  donde el número de accidentes sea mayor que 200?

cur.execute("""
    SELECT Sales, SUM(Work_accident)
    FROM Detalle
    GROUP BY Sales
    HAVING SUM(Work_accident) > 200;
""")
res = cur.fetchall()
res
```

Out[15]:
```
[('sales', 587.0), ('support', 345.0), ('technical', 381.0)]
```